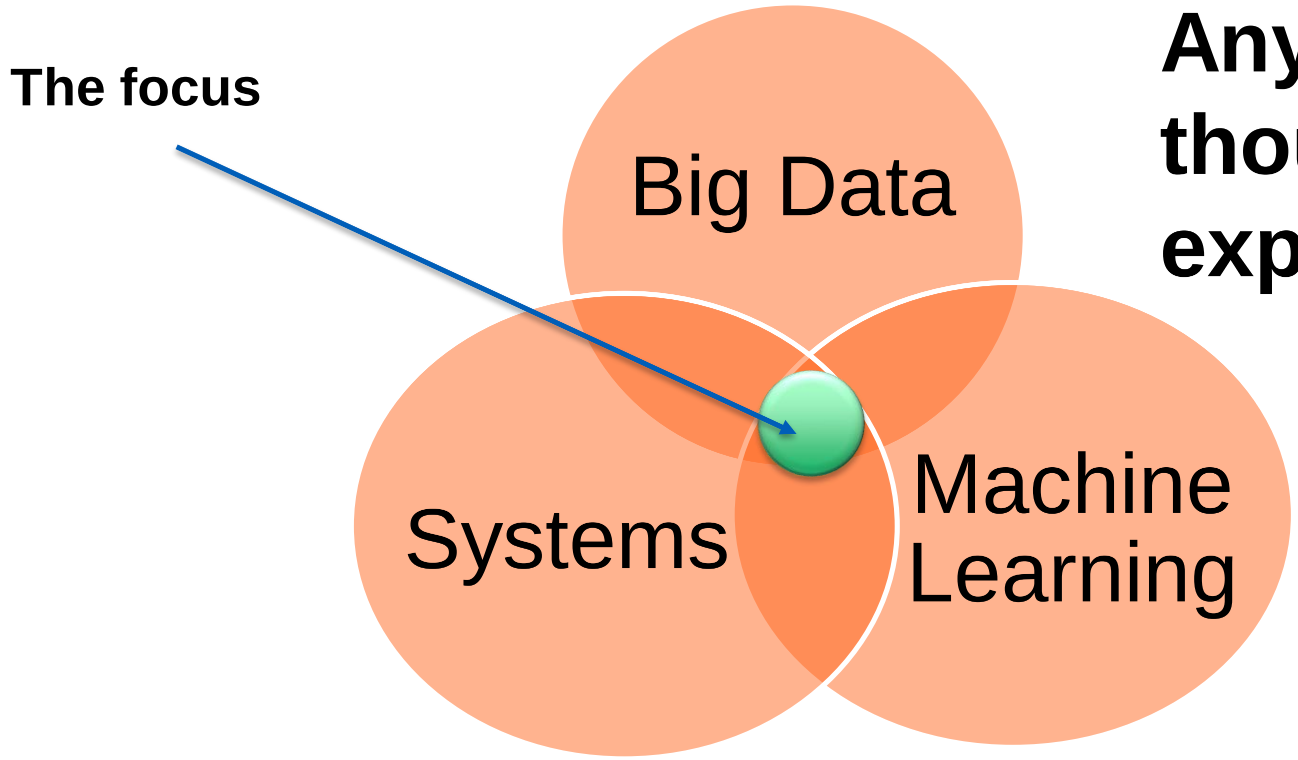
Aalto University
School of Science

Coordination of Big Data/ML Tasks

Hong-Linh Truong
Department of Computer Science
linh.truong@aalto.fi, *https://rdsea.github.io*

Our focus in this course



Any idea, thought, expectation?

Content

- **Pipeline coordination**
  - Orchestration style
  - Choreography style
- **ML Model Serving**
- **Experiment Management**
- **Study log**

Pipeline coordination

Examples of Requirements

Discussion:

- **ML Phases & Tasks**
- **Software stack**
- **Execution environments**
  - Computing resources
- **R3E**

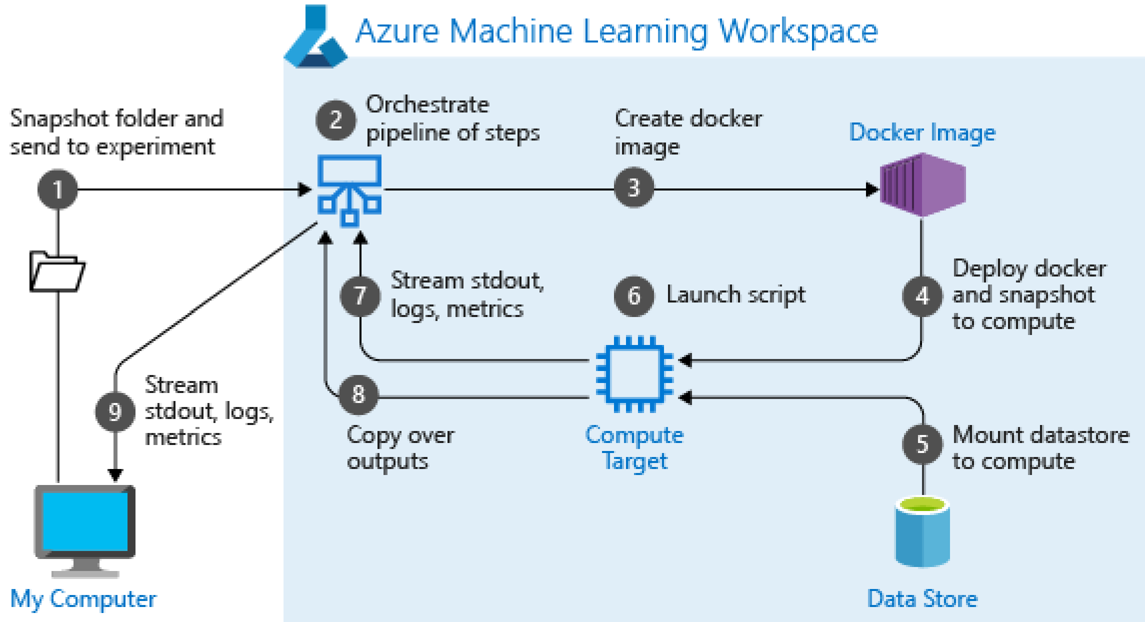
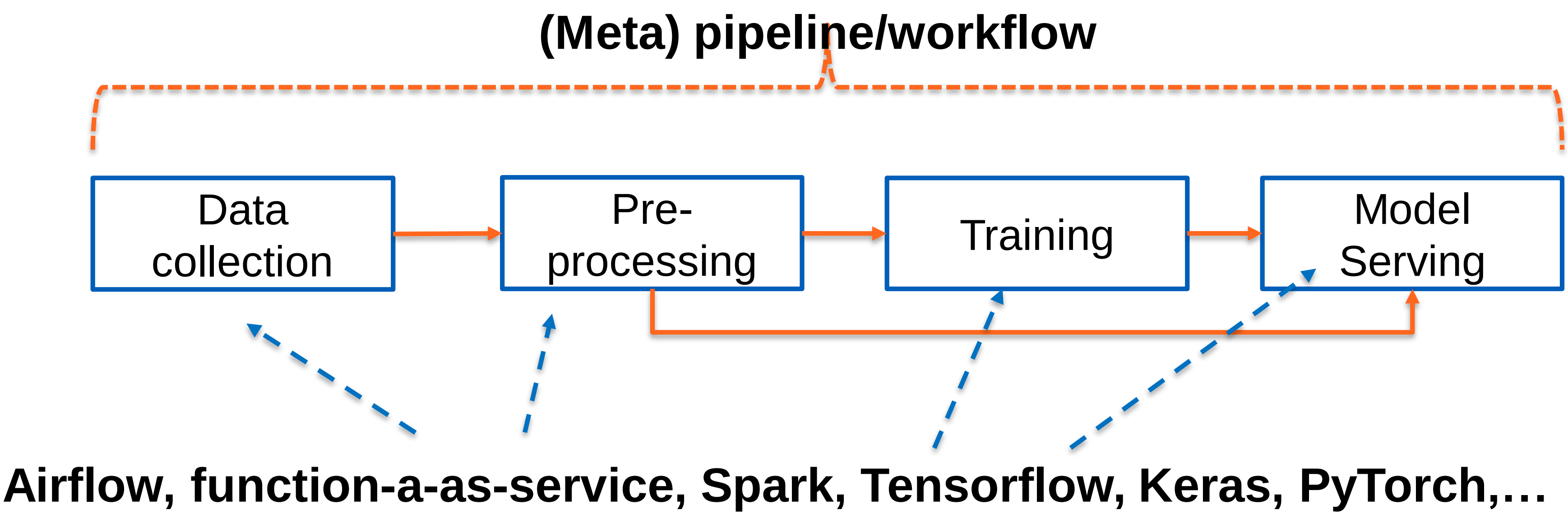


Figure source: https://docs.microsoft.com/en-us/azure/machine-learning/concept-ml-pipelines

The pipeline view of big data/ML systems

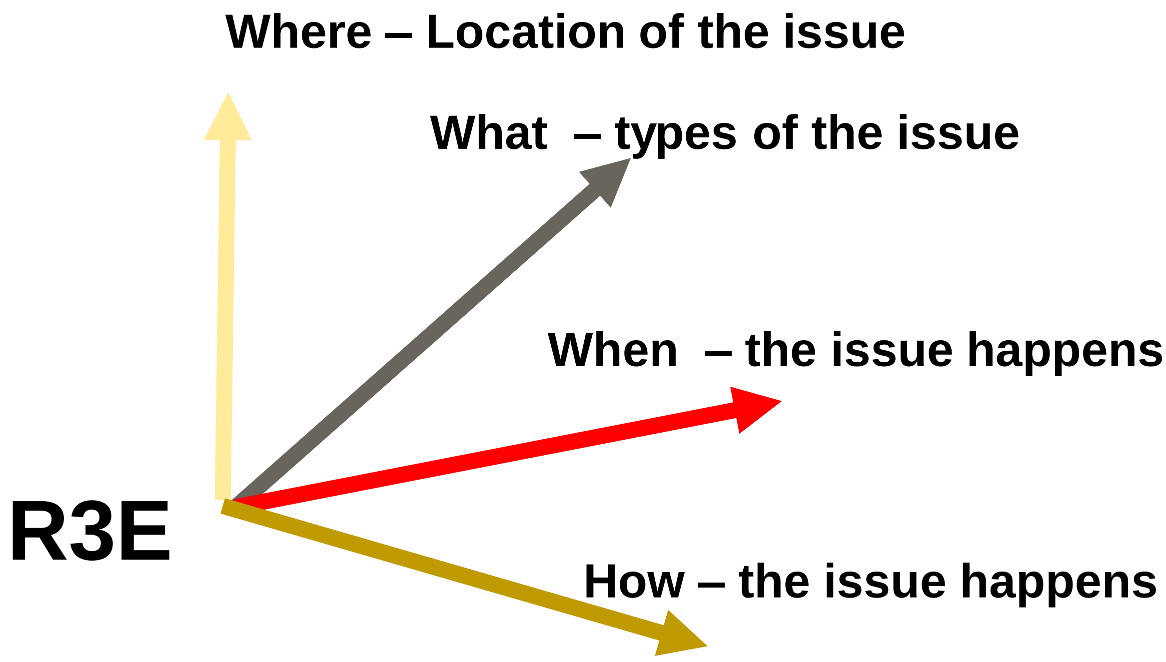
- **Multiple levels:**
  - Meta-workflow or -pipeline
  - Inside each phase: pipeline/workflow or other types of programs



Main issues related to coordination

- **How to coordinate phases and tasks in big data/ML systems**
  - Automation is an important requirement
- **How to assure R3E for the pipeline execution**
  - End-to-end R3E requires coordination
  - Issues in internal and external services
- **How to manage experiment data**
  - Trial computing configurations
  - Inputs/results

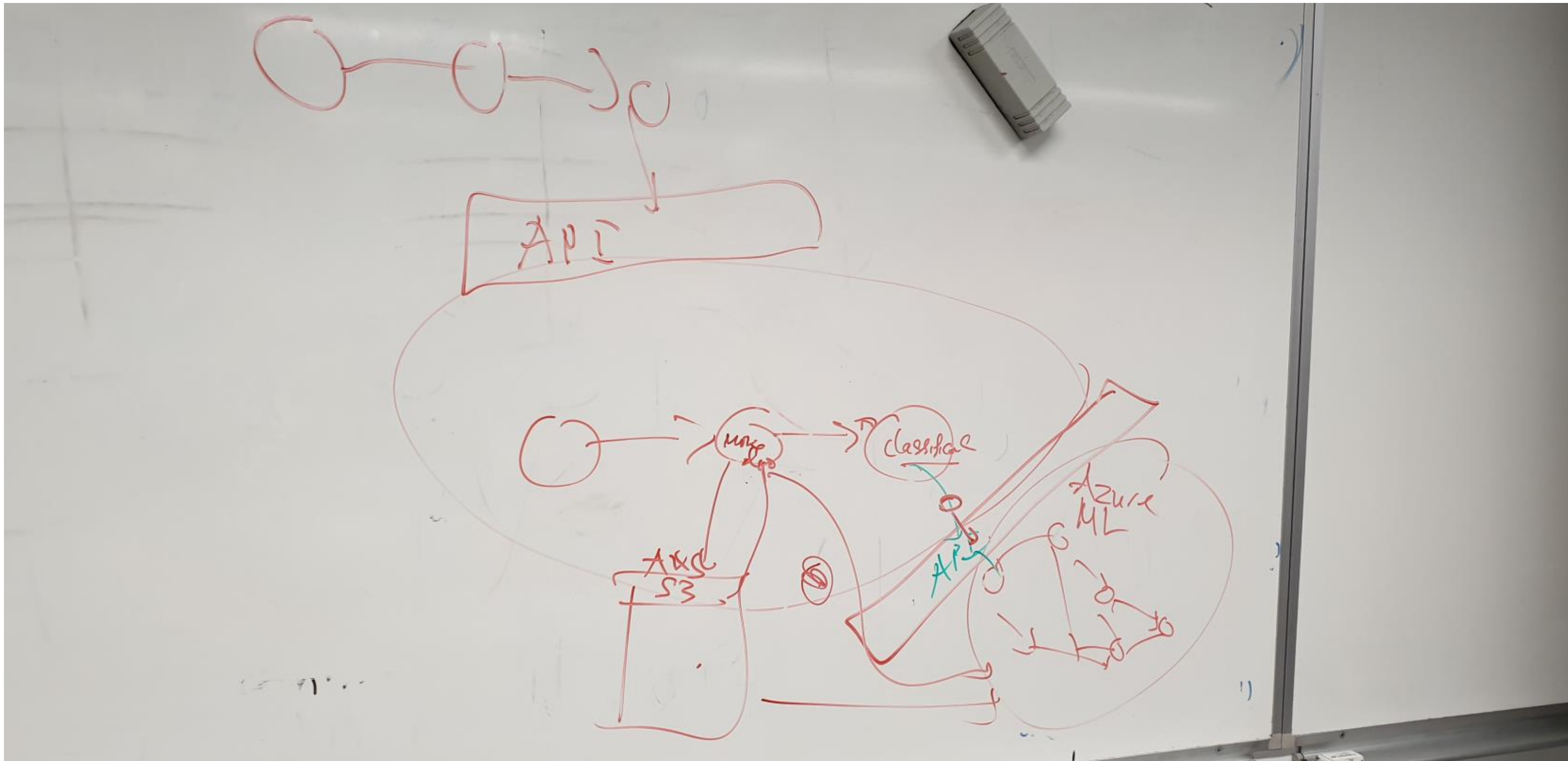
W3H: what, when, where and how for R3E issues



Key notions

- **Workflow and Task/Activity/Step**
- **Important notes on the abstraction**
  - A task can encapsulate a “complex workflow”
- **Software frameworks**
- **Platform services**
  - Services offering features/functionality for executing “tasks”
  - Single or multiple the providers?
- **Execution environments and resources**
  - Single platform or cross (heterogeneous) platforms

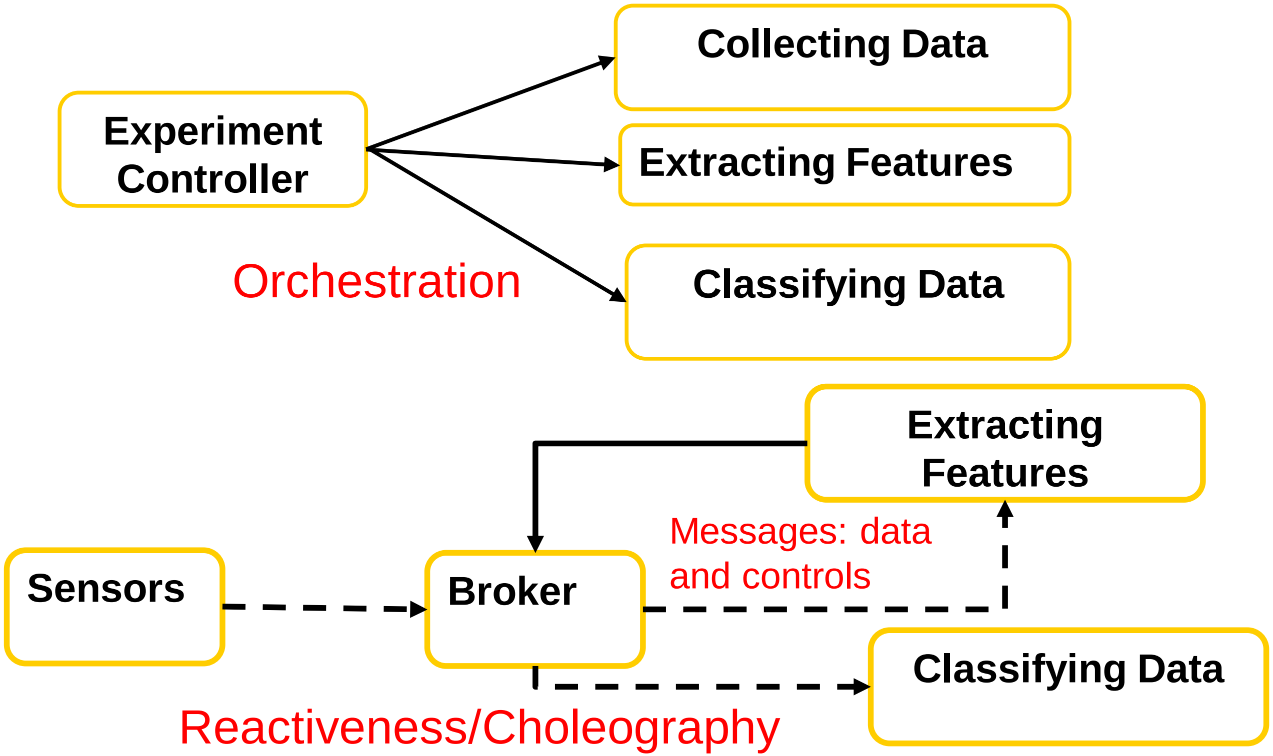
A task encapsulating a workflow via API



Coordination Styles

- **Coordination models for Big Data/ML systems**
  - Orchestration and reactiveness/choreography
- **Orchestration**
  - Task graphs and dependencies are based on control or data
  - Triggered based on completeness of tasks or the availability of data
- **Reactiveness/Choreography**
  - Follow reactive model: tasks are reacted/triggered based on messages

Orchestration and Reactiveness



System issues impacting R3E and coordination

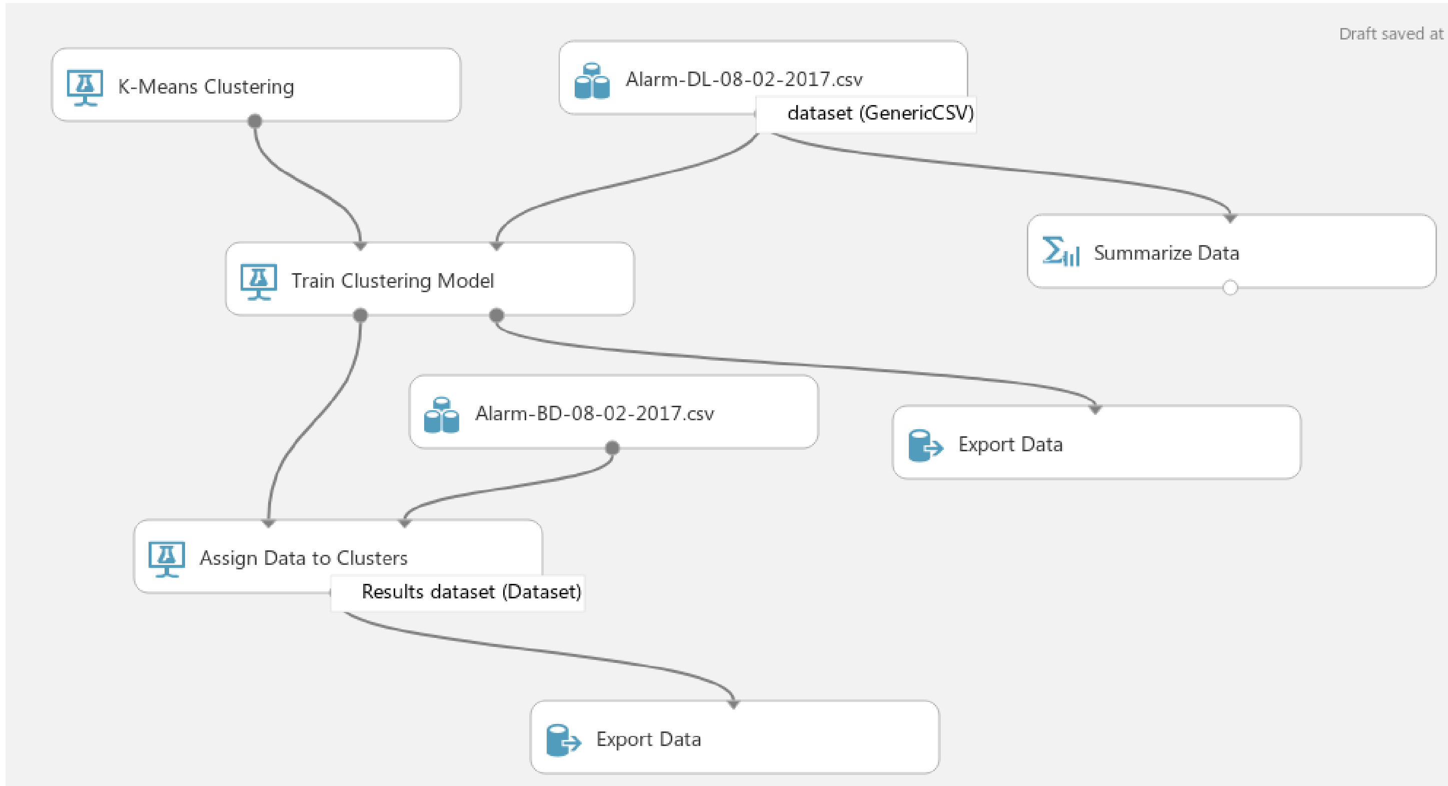
- **Main situations:**
  - Within the same system/infrastructure
    - *All services and computing resources belong to the same platform/infrastructure*
    - *E.g., running everything with Google Cloud or Microsoft Azure*
  - Across systems/infrastructures
    - *Services in different clouds or cloud data centers*
    - *E.g., Edge-cloud infrastructures*
  - The same software stack or not?
- **How such situations would affect the coordination/R3E?**

Workflows

- **Examples like**
  - Apache Airflow, Azure ML Pipeline
- **Often running in the same infrastructure**
- **Task-driven or data-driven specification**
- **Generic workflows**
  - Use to implement different tasks, such as machine provisioning, service calls, data retrieval
    - *Examples: Airflow, Argo Workflows*
- **Specific workflows for specific purposes**
  - E.g., Kubeflow (https://github.com/kubeflow/pipelines)

Workflow used in ML pipelines

Azure ML Pipelines



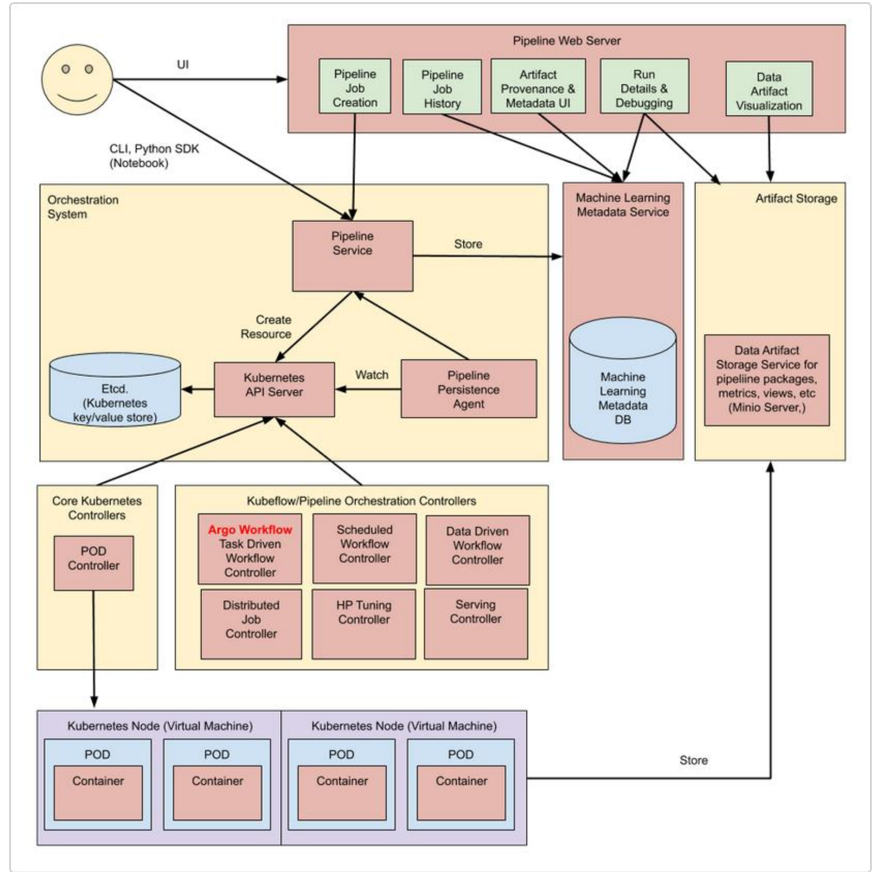
Orchestration architectural style: design

- **Workflow architectures are known**
  - Big Data/ML systems: leverage many types of services and cloud technologies
- **Required components**
  - Workflow/Pipeline specifications/languages (also UI)
  - Data and computing resource management
  - Orchestration engines (with different types of schedulers)
- **Execution environments**
  - Cloud platforms (e.g., VMs, containers, Kubernetes)
  - External services

Examples: Kubeflows

- **End-to-end Orchestration**
- **Orchestration is based on workflows**
- **Using "Orchestration controllers"**

- Discussion: dealing R3E with ML workflows?
  - Where, What, When and How



https://www.kubeflow.org/docs/pipelines/overview/pipelines-overview/

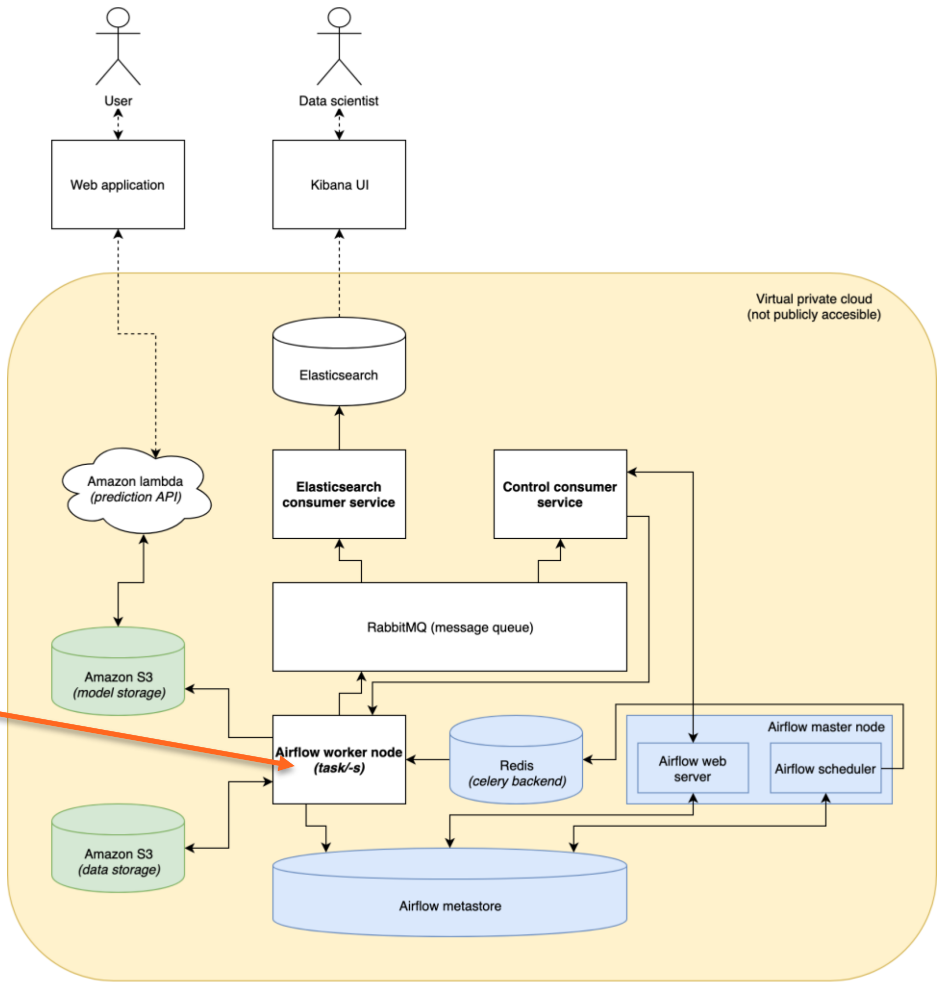
Examples: Coordinating tasks

- Discussion: dealing R3E with ML workflows?
  - Where, What, When and How

Source: Kreics Krists, „Quality of analytics management of data pipelines for retail forecasting,“, Aalto CS Master thesis, 2019

Examples: Exchanging metrics for R3E coordination

Monitoring various metrics, including user-defined quality of data

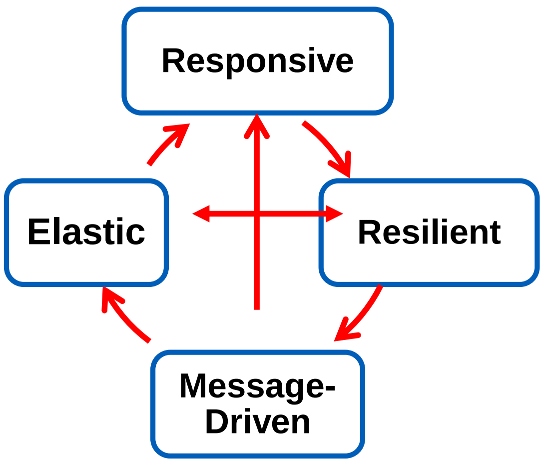


Source: Kreics Krists, „Quality of analytics management of data pipelines for retail forecasting“, Aalto CS Master thesis, 2019

Choreography: Reactive systems for Big Data/ML

Do you remember key principles of reactive systems?

Reactive systems



Source: https://www.reactivemanifesto.org/

- Responsive: quality of services
- Resilient: deal within failures
- Elastic: deal with different workload and quality of analytics
- Message-driven: allow loosely coupling, isolation, asynchronous

Reactive systems for Big Data/ML: methods

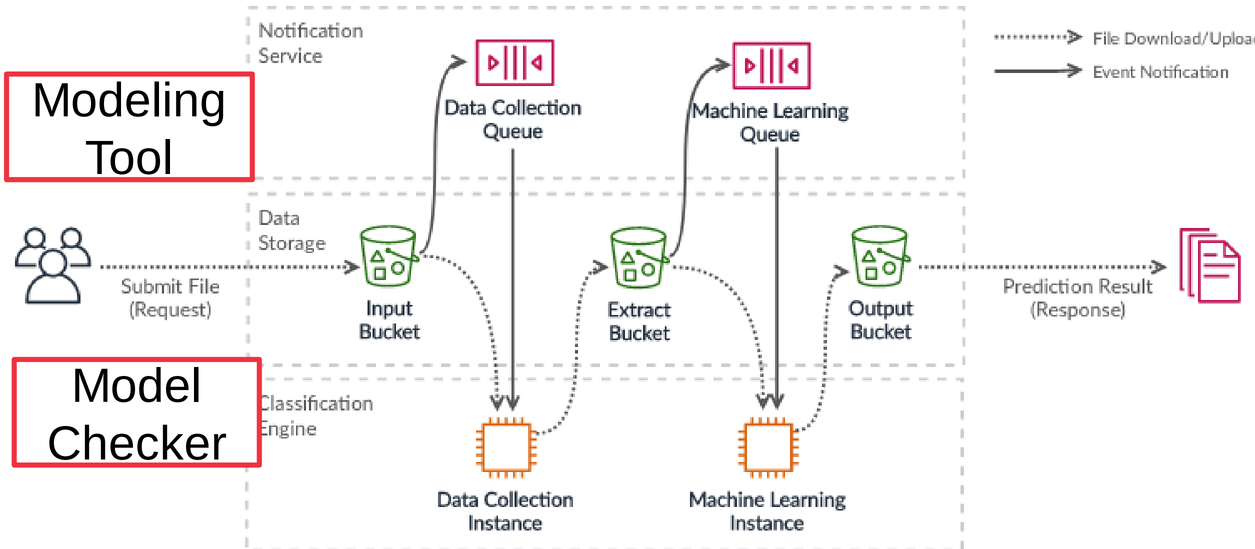
- **Have different components as services**
  - Components can come from different software stacks
- **Elastic computing platforms**
  - Platforms should be deployed on-demand in an easy way
- **Using messages to trigger tasks carried out by services**
  - Messages for controls and for data
  - Heavily relying on message brokers and serverless (function-as-a-service)

Examples: do-it-your-self ML classification for BIM

- Discussion: dealing R3E with ML workflows?
  - Where, What, When and How

Modeling Tool

Model Checker



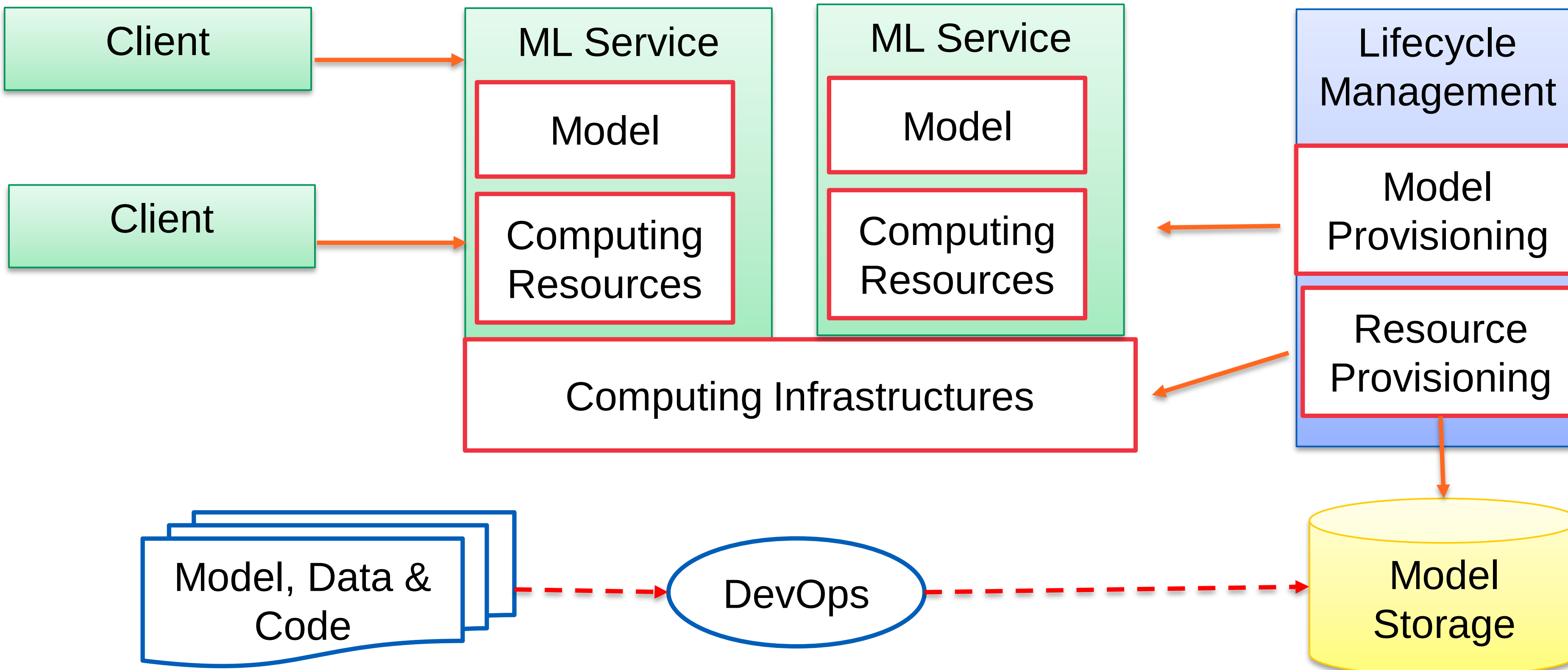
Source: Minjung Ryu, „Machine Learning-based Classification System for Building Information Models “, Aalto CS Master thesis, 2020

Dynamic ML Serving

ML Model Serving

- **Allow different versions of ML models to be provisioned**
  - Runtime deployment/provisioning of models
  - “Model as code” → can be deployed into a hosting environment
- **Why? Anything related to R3E?**
  - Concurrent deployments with different “pay-per-use” principles (elasticity as well?)
  - A/B testing and continuous delivery for ML (https://martinfowler.com/articles/cd4ml.html)
- **Existing platforms**
  - Increasingly support by different vendors as a concept of “AI asa service” (check https://github.com/EthicalML/awesome-production-machine-learning#model-deployment-and-orchestration-frameworks)

ML Model Serving design



Example: TensorFlow Extended Serving

- **Lifecycle**
  - Load, serving and unloading
- **Metrics & Policies**

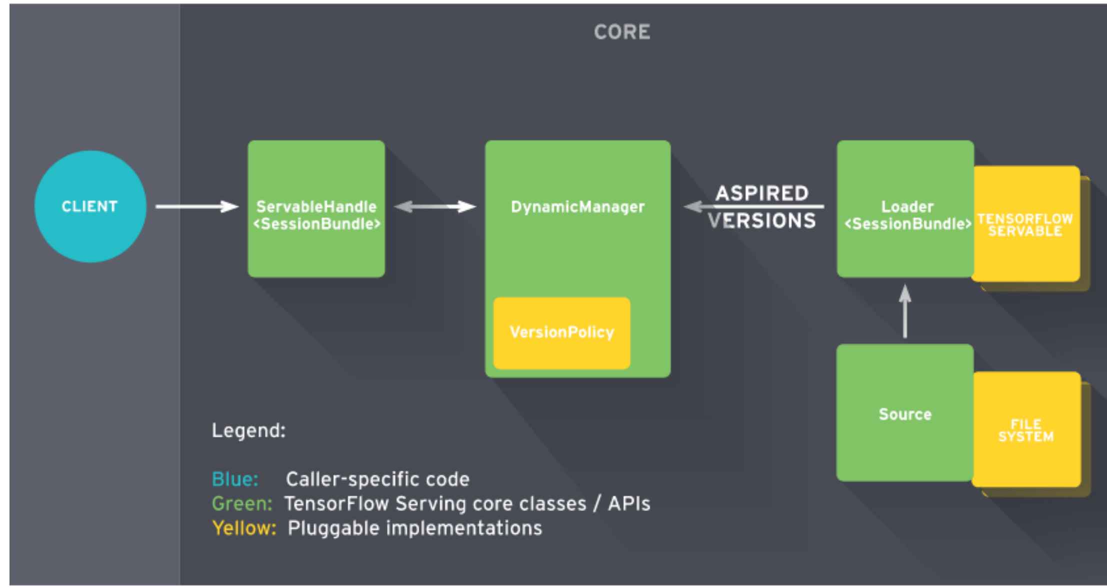
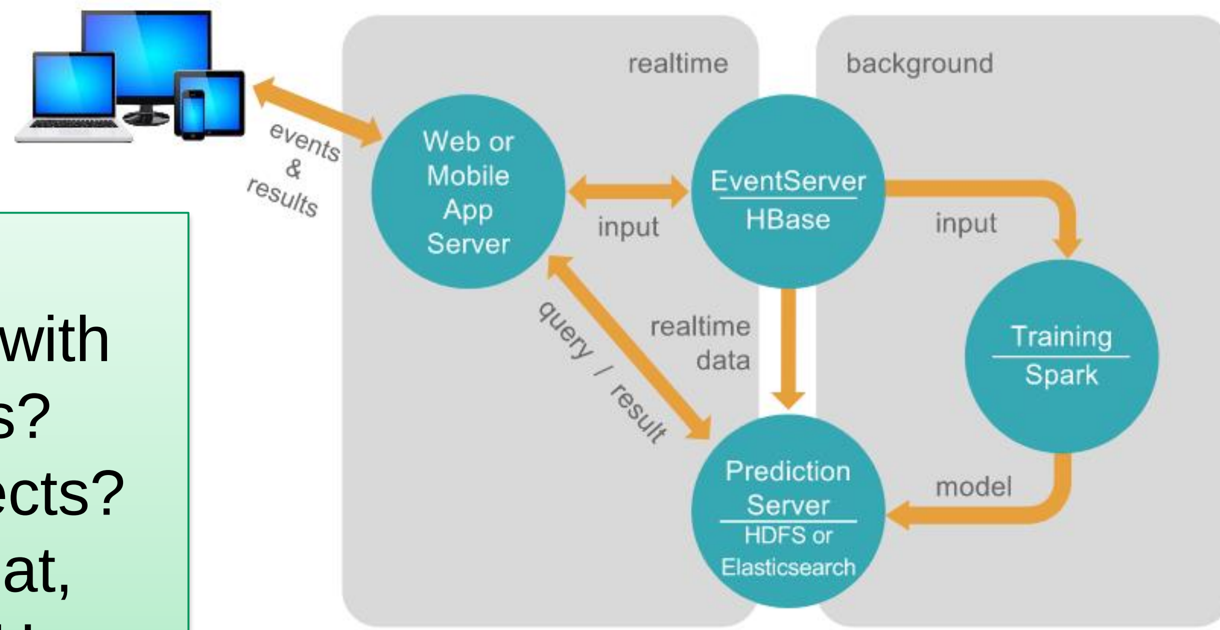


Figure source: https://www.tensorflow.org/tfx/serving/architecture

Example of Prediction.io



- Discussion: dealing R3E with ML workflows?
  - Elastic objects?
  - Where, What, When and How

Figure source: https://predictionio.apache.org/system/

Experiment management

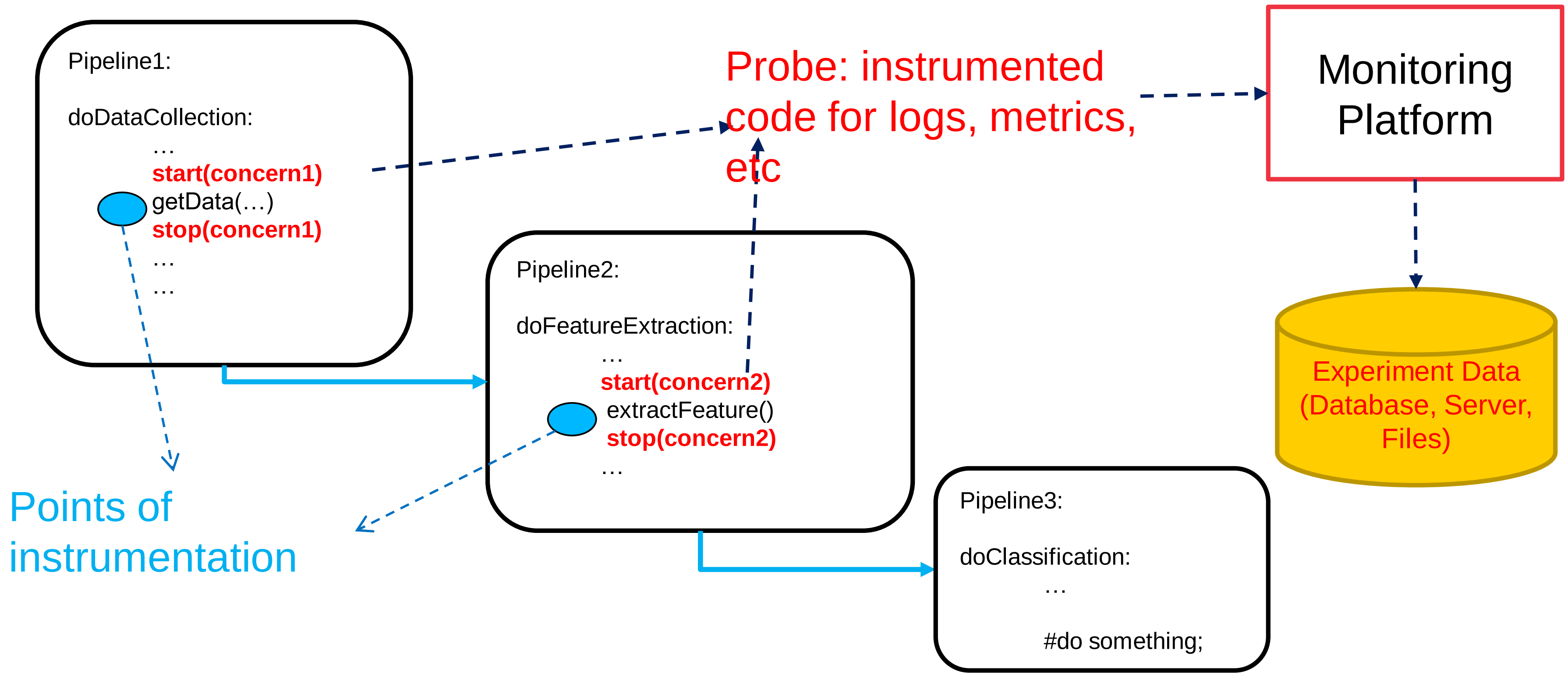
Problems

- **We need to run many experiments**
  - Known domain and well-known books (e.g., "Design and Analysis of Experiments" by Douglas C. Montgomery)
  - How does it work in ML?
- **Coordinating runs in experiments is also an important task**
  - How does this help to understand and support R3E?
- **What do we need?**
  - Tools/frameworks for tracking experiments

Notions

- **A single run/trial**
  - Inputs, results, required software artefacts
  - Computing resources, logs/metrics
- **Experiment**
  - A collection of runs/trials/executions gathered in a specific context
- **Steps**
  - Parameterization: generate different parameters
  - Deployment: prepare suitable environments
  - Execution: run and collect metrics
  - Analysis and Sharing: analyze experiment data

Experiment tracking



But remember it is very large system! Which tools can we use?

Examples

- **Experiment in Azure ML SDK**
  - https://docs.microsoft.com/en-us/python/api/overview/azure/ml/?view=azure-ml-py#experiment
- **MLFlows**
  - https://mlflow.org/
- **Kubeflows**
  - https://www.kubeflow.org/docs/pipelines/overview/concepts/

Examples: MLFlow APIs

- **Experiment**

`mflow.start_run()/end_run()`

- **Logs/metrics collection**

`mflow.set_tag()`

`mflow.log_*()`

- **Tracking data management**
  - Local files, Databases, HTTP server, Databrick logs

Study log

P1 - Take one of the following aspects:

- Robustness, Reliability, Resilience or Elasticity

P2 - Check one of the following aspects:

- Orchestration, ML model serving or Experiment Management

In a *specific software framework* (F3) that you find interesting/relevant to your work:

discuss how do you see F3 supports P1 in doing P2

Thanks!

Hong-Linh Truong
Department of Computer Science

rdsea.github.io